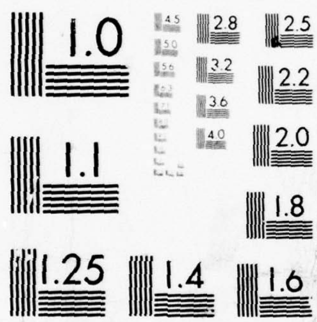
AD-A071 203 DREXEL UNIV PHILADELPHIA PA DEPT OF MECHANICAL ENGIN--ETC F/G 9/2
MICROPROCESSOR UTILIZATION IN ULTRASONIC NONDESTRUCTIVE INSPECT--ETC(U)
JUN 79 J L ROSE, G H THOMAS N00014-77-C-0607

UNCLASSIFIED NL

1 OF 1
AD
A071203

END
DATE
FILMED
8--79
DDC




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

9 Technical rept.,

6 Microprocessor Utilization in Ultrasonic Nondestructive Inspection Systems.

10 Joseph L. Rose
Professor of ~~Mechanical~~ Engineering & Mechanics

~~and~~

Graham H. Thomas
Graduate Student

Drexel University
Philadelphia, Pennsylvania 19104



Sponsored by

15 Office of Naval Research
Contract N00014-77-C-0607

11 15 June 1979

12 38p.



390 782 GM

Presented at the 1978 Fall ASNT Conference in Denver, Colorado.

## Acknowledgments

The authors would like to thank the Office of Naval Research under the guidance of Dr. Nicholas Perrone for their support of this project. Also, the authors would like to thank Krautkramer-Branson, Inc. for their computer support.

## Abstract

The purpose of this work is to demonstrate the feasibility of using microprocessors in ultrasonic inspection. Benefits of employing a microprocessor are two-fold, the first with respect to the development of a small, compact and portable inspection device, and the second with respect to obtaining flaw data in digital format for use in structural mechanics computer programs for stress analysis. An ultrasonic problem was selected that clearly demonstrates the capability of a microprocessor with initial attention on problem simplicity from an ultrasonic inspection point of view. Such items as data storage, computational speed, computer language selection and utilization, interfacing techniques and the use of such peripheral devices as analog to digital converters and display devices are reviewed in the paper.

The problem selected for this study is that of modelling an ultrasonic inspection of the steel plates from the hull of a ship. The procedure will be applicable to both quality control during manufacture and in-service inspection. Procedures for implementing an advanced ultrasonic inspection algorithm applicable to adhesive bond inspection is also outlined in the paper.

## Introduction

The use of high speed digital computers in ultrasonic inspection has become tremendously popular as the science of ultrasonics becomes more sophisticated. Most of the more difficult ultrasonic inspection problems are first considered in the laboratory in a variety of feasibility experiments. The laboratory provides us with the computational efficiency of large computers, plus an ideal controlled environment to expedite the solution to difficult problems. Once a solution has been developed, the next step is to adapt the technique to the field situation. One of the major difficulties in field implementation is the unsuitability of the large high-speed computers for field inspection. Thus, one key to field implementation success is the utilization of a microprocessor, which replaces the laboratory based computer, the microprocessor approach serves as an alternative to the hardware analog circuit system which is often limited in its total capability of implementing advanced computational algorithms for inspection and reflector classification.

Such items as data storage, computational speed, computer language selection and utilization, interfacing techniques and the use of such peripheral devices as analog to digital converters and display devices are reviewed in the paper.

One purpose of this paper is, therefore, to demonstrate the feasibility of incorporating a microprocessor into an actual ultrasonic inspection system. An LSI-11 microprocessor was used as the central processing unit for an accurate and fast C-scan system for flaw growth determination. A sample inspection problem for the U.S. Navy was considered in this work, in particular that of inspecting large plates from ship hulls. Mapping procedures for damaged areas are presented for both quality control and in-service inspection problems.

To use this microprocessor based inspection system developed by this study, the area of interest could first be determined either from a quality control examination, a possible acoustic emission, liquid penetrant, or visual inspection. Then the proposed system is used to manually scan the area of interest, record the location of the edge of the damage and present a complete picture of the extent of the damage. In addition, this equipment is also designed to monitor the damaged area and to note any growth while the ship is in service. Benefits of employing a microprocessor are two-fold, the first with respect to the development of a small, compact and portable inspection device, and the second with respect to obtaining flaw data in digital format for use in structural mechanics computer programs for stress analysis. Once the extent of the flaw is determined, for example, the information can be incorporated into a structural dynamics finite element code to calculate the criticality of the damage.

Once the ship hull inspection problem was solved, the potential of the microprocessor for ultrasonic inspections was further explored. The difficult problem of nondestructively determining adhesive bond strength was considered for implementation on the microprocessor system. Procedure for implementing an advanced ultrasonic inspection algorithm applicable to adhesive bond inspection is also outlined in the paper.

## Microprocessor Selection Considerations

A most critical step in developing an ultrasonic data acquisition and signal processing system is the selection of the central processing unit, in this case a microprocessor. Since this study involved only small portable computing packages, the types of central processing units (CPU) can be separated into dedicated microprocessors and programmable microcomputers [1]. The dedicated system is permanently programmed to perform a specific algorithm only. Usually the program is developed on a larger computer system and "burned" into the memory of the CPU. In general, as long as the microprocessor can handle the instruction set of the program and has sufficient memory size to hold the program, then it is acceptable. The only other considerations in the dedicated system would be physical such as size, weight, and power consumption. A more flexible, programmable, microprocessing system requires many other considerations before selecting the optimal machine.

The programmable microprocessor is basically a scaled down version of a traditional computer. The microprocessor's internal architecture is similar to the large computer, only the microprocessor relies on large scale integration (LSI) and other sophisticated electronics to shrink its size [1]. The first consideration when choosing a microprocessor is the word length. The word length is the number of bits that the computer can process as a single, primary unit. A larger word length has many advantages. A large word length allows the computer to handle big numbers easily. Thus the word size is directly proportional to the size of the instruction set, the number of memory locations that can be addressed, and the accuracy of the mathematical calculations. The word size is also somewhat responsible for the speed of the machine. For example, a 16 bit machine could handle an arithmatic

addition in one step whereas an 8 bit machine might need two or more steps to do the same addition. The number of memory addresses and therefore the size of usable memory is dependent on the word length. Memory size is important for loading advanced algorithms and larger memory sizes are needed to use higher level languages such as Fortran. Also the longer word length allows more significant digits to be used during arithmatic calculations and thus provides better accuracy.

Another important consideration when selecting a programmable microprocessor system is its flexibility for interfacing an assortment of peripheral devices. This flexibility depends strongly on the bus architecture. The bus allows the various computer components such as the CPU and interface cards for peripheral devices to be connected for transferring information. A convenient bus structure incorporates asynchronous, bidirectional communications. Asynchronous means that each peripheral device whether serial or parallel , can pass data along the bus at the device's own speed and thus no special interface is needed. The bidirectional characteristic of a bus means that information is passed in both directions, between CPU and peripheral on the same set of lines, this provides a simpler and more easily programmed bus structure. The most flexible microprocessor system would handle parallel, serial and direct memory access (DMA) interfacing. Parallel data transfer means that the data is passed a word at a time which necessitates a transfer line containing at least as many conductors as there are bits in the word. This method of data transfer is the quickest and is the most common. Serial communication was the first invented and is better suited to sending data over longer distances. The serial method uses basically a single conductor to send the information one bit after another. An interface card must then

convert the string of data bits into words and pass these words in parallel form to the CPU. A universial asynchronous receiver transmitter (UART) is the primary component in a serial interface for parallel formating.

A final consideration for microprocessor selection is speed of computation. The microprocessor is inherently slower than the larger computers, but this difference is quickly diminishing. There are several characteristics of the microprocessor which influence its speed. The first characteristic is the electronic component design which includes the CPU's internal clock and the bus cycle time. This sets the rate for instruction processing and data transfer. A valuable characteristic of some microprocessor is direct memory access (DMA) which allows the direct transfer of data from a peripheral device to the computer's memory without CPU intervention. This form of data transfer does not depend on bus cycle time and therefore is a much faster method to transfer data. And finally, the software influences the computation rate. Initially microprocessor had limited memory size and they had to be programmed in a machine language, but as the microprocessor advanced so did the assortment of programming modes.

## Programming Considerations

A primary consideration in the development of software for the microprocessor is the language to be used. Microprocessors in general, due to their limited internal storage must be programmed with an efficient language [2,3]. The most basic of computer languages is called machine language which consists of a list of numbers. It is the machine language that the computer actually uses to perform its operations. A list of numbers is difficult for a person to understand, so a more advanced language was designed that used mnemonics for instructions. This language is called assembly language and is easily translated into machine language. Higher level languages, such as Fortran, consist of English words for instructions and need a sophisticated assembler to translate the words into machine language. This assembler requires a large amount of internal storage and does not translate the program into the most efficient form. Higher level languages are designed to run on large computers where internal storage is not a limiting factor. But when programming a microprocessor, the ease of programming in Fortran must be sacrificed for the efficiency of assembly language.

A study was conducted as part of this report to determine the optimal programming mode for the LSI-11 microprocessor. Since the LSI-11 microprocessor is compatible with the PDP11/05 minicomputer (see Fig. 1), the software was developed on the minicomputer and loaded into the microprocessor. A unique feature of the LSI-11 microprocessor is its large internal storage which allows for some higher level language usage. A balance of higher level language and assembly language would be best. Certain device handlers and time consuming routines should be programmed in assembly language while the main program should be written in Fortran to decrease development time. An example of the time savings of assembly language was demonstrated by a comparison of a fast Fourier transform (FFT) in Fortran and a FFT in assembly language. The Fortran version took 33 seconds to compute the Fourier

transform of a 256 point amplitude-time signal, and the assembly language program took 3 seconds to transform the same signal. Therefore, the assembly language FFT could save considerable amounts of time in a program that calculated many transforms. The higher level language is used in the main program because the routines are not time consuming and the high level language, Fortran, is much easier to write. The reason Fortran is not as fast or as efficient as assembly language is because a single Fortran statement is actually translated by a compiler into generally several machine language steps. The way the compiler does this translation is not always the best or most efficient and this more time is consumed executing that statement as compared with an assembly language statement which is usually a one for one translation to machine language (a binary number) and thus does not involve a compiler. Also the assembly language programmer must tell the computer explicitly how to run the program and thus can specify the most efficient manner. This means that the assembly language program takes many more statements than the Fortran counterpart and is usually more difficult to develop. A Fortran listing and an assembly language listing of a program that multiplies two numbers is displayed in Figures 2 and 3, respectively. These listings illustrate the difference in the number of programming steps and complication between Fortran and assembly languages. The combination of faster, more efficient assembly language subroutines and easily developed higher level language main programs provide a powerful, versatile software development system.

Sample problem

The problem of scanning a large steel plate and determining the extent of any damage was divided into four segments. First, a mechanical scanner was designed and built to record accurately the location of the transducer on the plate. The scanner, as shown in Figures 4 and 5, is attached to the plate at its pivot point by either a magnet or a suction cup. The transducer is attached to a slider which is mounted on an arm that is allowed to turn about the pivot point. A polar coordinate representation of the transducer's location is obtained from potentiometers, one located at the pivot which provides the angle, and one attached to the slider by a string and pulley system which provides the radius.

The second part of the system is the data acquisition equipment which is shown in Figures 5 and 6, and consists of a USIP-11 flaw detector with peak detector and an inexpensive analog to digital converter. The USIP-11 is used to pulse the transducer and to gate out the significant signal. Its peak detector will determine the peak voltage of the gated signal and the analog to digital converter will digitize that voltage so it can be stored in the microprocessor. Therefore, each data point will be an element of an array which will be stored in the computer in the form of a radius, an angle, and a voltage ($r$, $\theta$, $P$).

The third part of this inspection system is the signal processor. The signal processor consists of an LSI-11 microprocessor, a video display terminal, a paper tape reader/punch and the necessary interfacing hardware, Fig. 7. An LSI-11 microprocessor system was used for studying such problems as storage requirements, computation time, interfacing problems, programming modes and general adaptability to nondestructive testing kinds of problems. The LSI-11 unit was selected for several reasons, the principle reason of

which is associated with the availability of a development system in our laboratory, namely the PDP11/05 minicomputer system. Also, the LSI-11 is one of the more versatile microprocessors with more than sufficient memory capacity for this problem. This signal processing system must store all the data points and their respective voltage levels, plus determine which data points locate the edge of the damaged area. Once the edge points have been separated, a picture of the damaged area can be displayed on the video terminal and punched on paper tape for a permanent record.

The fourth and final segment of this project was to design and construct test plates for determining the accuracy of the system. Two ship hull model specimens have been designed to test the capabilities of the microprocessor based, ultrasonic C-scan inspection system. Drawings of the two test specimens are shown in Figures 8 and 9. These specimens were machined from 1/2" thick steel plates and included various curvatures, right angles, and depths. The manual C-scan apparatus was used to determine the location and shape of each flaw. An account of the accuracy and reproducibility of the microprocessor based C-scan system was gained by using these two models.

## Results

The utility of a microprocessor in an ultrasonic inspection system is certainly demonstrated with the design and implementation of this microprocessor augmented C-scan system. There are two options for the form of the output from this microprocessor based ultrasonic test apparatus. The output may be displayed on the video terminal by a crude graphics mode. An example of this display is shown in Figure 10. Also, the microprocessor is able to punch on paper tape the precise coordinates sent to the microprocessor from the potentiometers of the C-scan device. The precise coordinates can additionally be plotted on a more sophisticated terminal if the microprocessor is directly interfaced, for example, to a PDP-11 terminal. The graphics package on the PDP-11 terminal was used to demonstrate the accuracy of the microprocessor controlled C-scan system. PDP-11 drawn plots are shown in Figures 11 and 12 along with an actual drawing of the flaw.

## Advanced Ultrasonic Implementation Problem

A large number of problems in nondestructive testing require the use of advanced techniques in pattern recognition. An outline of a sample problem for microprocessor implementation of an advanced ultrasonic inspection problem, that of adhesive bond strength prediction, is presented in the following text. The problem of adhesive bond inspection is of particular importance to the joint technology because adhesives are gaining wide usage in structures where weight savings, good load transfer properties, and environmental degradation resistance are important.

The problem of predicting adhesive bond strength ultrasonically has been studied by Rose and Thomas [4]. Their technique has been proven 91% reliable using a Fisher Linear Discriminant function to classify bond specimens into two classes, good and bad. For industry to take full advantage of such a technique, a microprocessor based ultrasonic data acquisition and signal processing system is needed. This system would provide the portability necessary for in the field inspections of bonded structures and also provide the computing power necessary to operate the data acquisition equipment and run the advanced pattern recognition algorithm.

The components needed to assemble an adhesive bond inspection system are listed in table 1. This system is controlled by an LSI-11 microprocessor with 28K words of memory. The microprocessor must be interfaced with a high speed analog to digital converter comparable to the the Biomation 8100. The microprocessor must be capable of operating the sophisticated device handler used to control the A/D converter. A mass storage device is optional but is useful for storing data and developing software. A video terminal is used for communicating with the microprocessor and for displaying the results of the inspection. The data acquisition is done by the ultrasonic equipment that includes an ultrasonic pulser/receiver, a 20 MHz broadband contact transducer, and an oscilloscope to monitor the R-F wave-

form. Figure 13 is a block diagram of the ultrasonic, adhesive bond inspection system. For an operational system the microprocessor must be able to handle the following software. This software must be contained in the microprocessor's internal memory if a mass storage device, such as floppy disk, is not included. First, the device handler and corresponding subroutine to control the A/D converter is needed. Then the programs or subroutines to reduce the data and extract the features are added in. And finally the actual pattern recognition algorithm, in this example the Fisher Linear Discriminant function, is included to predict the adhesive bond strength. The LSI-11 microprocessor studied for this project had 28K words of memory which is sufficient to hold the necessary software to perform the adhesive bond strength prediction algorithm. The system is not actually running advanced ultrasonic algorithms because the Biomation 8100 analog to digital converter has not yet been interfaced to the microprocessor. Instead, the microprocessor was simulated on the PDP-11/05 minicomputer which has the A/D converter interface. The PDP-11/05 minicomputer has the same internal memory as the LSI-11 microprocessor plus the operating instruction set for both machines is identical. If the advanced ultrasonic inspection program can run and fit in the memory of the minicomputer, it can also be handled by the microprocessor. The main difference between the two computers is the computing time. The microprocessor is slightly slower but since the ultrasonic inspection process does not rely on "real-time" predictions, this speed difference is not deterring. In conclusion, an implementation of a microprocessor, such as the LSI-11, is definitely possible for advanced ultrasonic inspection algorithms.

## Software for Adhesive Bond Inspection

The adhesive bond strength prediction algorithm as developed on the PDP-11/05 minicomputer for simulation on the LSI-11 microprocessor is as follows. First, a subroutine is called to operate the Biomation 8100 analog to digital converter. This subroutine triggers the A/D converter and captures the ultrasonic R-F waveforms reflected by the adhesive bond layer. This ultrasonic signal is temporally averaged to reduce noise effects. Next, the amplitude-time pulse is transformed into the amplitude-frequency domain using a fast Fourier transform. The Fourier transform of a reference pulse is divided into the Fourier transform of the bondline echo to produce the transfer function of the bond layer [5]. Feature values are then determined from the amplitude-time signal, its Fourier transform and the transfer function. These features are listed in Table 2 and displayed in Figures 14, 15, and 16. Once the feature vector has been determined, the feature values are inserted into a Fisher Linear Discriminant function of the form:

$$a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_nx_n = b \quad [1].$$

The a's are the predetermined coefficients that when multiplied by the respective feature vector value (x) produce a scalar result, b. The b value is compared to a threshold value that is used to separate the two classes of bonds, good and bad. If the b value is less than the threshold, then the computer prints "bad spot" on the terminal and if the b value is greater than the threshold, the computer prints "good spot" on the terminal.

In summary, the results of the feasibility study using a PDP-11/05 minicomputer to simulate the LSI-11 microprocessor for implementation of an advanced ultrasonic inspection such as the adhesive bond problem clearly demonstrated the potential of a microprocessor in the ultrasonic inspection field.

## Conclusions

The microprocessor based, portable C-scan ultrasonic inspection system developed in this study is a significant advance in the area of ultrasonic field inspection. This study clearly indicates the usefulness of microprocessors for ultrasonic flaw detecting applications. Though there is still a tremendous amount of research needed to fully incorporate microprocessors in the field of ultrasonics, the potential has been realized by this work. The C-scan apparatus developed is an important addition to the ultrasonic field inspection technology. This system allows manual scanning of the area of interest, records the location of the edge of the damaged area and presents a graphical picture of the extent of the damage. The area of interest must first be determined either from a quality control examination, a possible acoustic emission, liquid penetrant, or visual inspection. In addition, this system is designed to monitor the damaged area and to note any growth while the ship is in service. The use of the ultrasonic C-scan device is not limited to ship hull inspection but for example, could be programmed to map delaminations in composite materials or even defective area in an adhesive bond layer. Therefore, the microprocessor in the C-scan system provide a great amount of flexibility in the system's applications.

A microprocessor based ultrasonic test system would also be used for solving advanced problems in adhesive bond inspection. As illustrated in this paper, the control of an analog to digital converter and the implementation of an advanced pattern recognition algorithm could be handled nicely with a microprocessor.

## References

1. Osborne, Adam, An Introduction to Microcomputers - Volume 1 - Basic Concepts, Adam Osborne and Associates, Inc. Berkeley, California, 1976.

2. Barden, William Jr., How to Program Microcomputers, Howard W. Sams and Company, Inc., Indianapolis, Indiana, 1977.

3. Eckhouse, Richard H. Jr., Minicomputer Systems: Organization and Programming (PDP-11), Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1975.

4. Rose, J. L. and Thomas, G. H., "Fisher Linear Discriminant Function for the Prediction of Adhesive Bond Strength," to be published in the British Journal of NDT, 1979.

5. Rose, J. L. and Avioli, M. J., "Transducer Compensation Concepts of Value in Flaw Classification," presented at the Spring ASNT Conference, New Orleans, 1978, and to be published in a reliability supplement to Materials Evaluation.

1) LSI-11 microprocessor with 28K words of memory

2) High speed analog to digital converter - comparable to Biomation 8100

3) Mass storage device - such as floppy disk

4) Video terminal for communications with microprocessor

5) Ultrasonic pulser/receiver

6) 20 Mhz broadband contact transducer

7) Oscilloscope to monitor ultrasonic R-F waveform

8) Auxiliary equipment such as connectors, cables, and interface cards

Table 1 - Components of a Microprocessor Based Adhesive Bond Inspection System

I. Amplitude Time Domain

1) PK/PK - peak to peak ratio of reference and echo signals

II Frequency Domain - Fourier Spectrum

1) FRQ. SHIFT - frequency shift between peak frequency of reference spectrum and peak frequency of echo from the bond layer's spectrum.

III. Transfer Function Domain

1) PK. FREQ. - peak frequency
2) DIP FREQ. - deepest depression frequency
3) DIP-PK FREQ - difference between frequency of dip and frequency of peak.
4) DIP/PK AMP - ratio of dip amplitude and peak amplitude
5) DP2-DP1 FREQ. - difference between frequency of secondary depression.
6) STD DEV. - standard deviation of transfer function
7) 6 DB BW - half way down bandwidth of primary depression

Table 2 - Features Used for Adhesive Bond Classification

Fig. 1- PDP-11/05 Minicomputer System

```
$
C        THIS PROGRAM TAKES TWO INTEGERS FROM
C        THE TERMINAL AND MULTPLIES THEM IN FORTRAN
         READ(5,1010)I
 1010    FORMAT(I5)
         READ(5,1020)J
 1020    FORMAT(I5)
         K=J*I
         WRITE(7,1030)K
 1030    FORMAT(I5)
         END
*
```

Fig. 2 - Fortran Listing of a Program to Multiply Two Numbers

```
$$/L$$
;       THIS PROGRAM READS TWO NUMBERS
;       FROM THE TERMINAL AND MULTIPLIES THEM
        .MCALL ..V2..,.REGDEF
        ..V2..
        .REGDEF
        .CSECT
        .TITLE TERM
TERM:   CLR     R0                      ;START MAIN
        JSR     5,READ                  ;GET CHARACTER
        MOV     R5,MULTI                        ;PUT CARACTER IN R1
        JSR     5,READ                  ;GET ANOTHER CHARACTER
        MOV     R5,R1                   ;PUT CHARACTER IN R1
        MUL     MULTI,R1                        ;ADD THEM UP
        JSR     7,DECURT                        ;PRINT THE ANSWER
MULTI:  .WORD   0
;       SUBR TITLE:  READ
;       THIS ROUTINE READS INTEGERS
;       FROM THE KEYBOARD
;       ADD SUBROUTINES TTIN AND TTOUT
READ:   CLR     NUM                     ;INITIALIZE OUTPUT
        CLR     COUNT                   ;INITIALIZE COUNT
        MOV     R5,SAVPC                ;SAVE THE PROGRAM COUNTER
        MOV     #BUFF,R4                ;SET UP BUFFER FOR CHARACTERS
GET:    JSR     7,TTIN                  ;READ CHAR
        MOV     R0,(R4)+                ;PUSH CHARACTER ON STACK
        INC     COUNT                   ;BUMP CHAR. COUNT
        JSR     7,TTOUT                 ;PRINT CHARACTER
        CMP     #15,R0                  ;LAST CHAR AN CR?
        BNE     GET                     ;NO, GET ANOTHER
        MOV     #12,R0                  ;YES, PRINT A LINE FEED
        JSR     7,TTOUT                 ;PRINT IT
        DEC     COUNT                   ;LOWER CHAR COUNT
        MOV     #BUFF,R4                ;SET UP BUFFER FOR READ
DB1:    MOV     (R4)+,R2                ;GET DIGIT
        SUB     #60,R2                  ;GET RID OF ASCII
        CMP     #11,R2                  ;IS IT LEGAL
        BLO     ERROR                   ;NO? DUMMY
        CMP     #6553,NUM               ;IS THE NUMBER TO BIG
        BHI     DB2                     ;NO
        CMP     #5,R2                   ;IS IT TO BIG
        BLO     ERROR                   ;YES? DUMMY
DB2:    ASL     NUM                     ;NUM*2
        ADD     NUM,R2                  ;NUM*2+DIGIT
        ASL     NUM                     ;NUM*4
        ASL     NUM                     ;NUM*8
        ADD     R2,NUM                  ;NUM=NUM*8+NUM*2+DIGIT
        DEC     COUNT                   ;LAST DIGIT
        BGT     DB1                     ;NO, DO SOME MORE
        MOV     NUM,-(SP)                       ;PUSH NUMBER ON THE STACK
        MOV     SAVPC,R5                ;GET RETURN ADRESS
        RTS     5                       ;GO BACK
ERROR:  .ASCIZ/*ERROR*/
SAVPC:  .WORD   0
COUNT:  .WORD   0
COLLUM: .WORD   0
NUM:    .WORD   0
BUFF:   .BLKW   12
;
;       SUBR TITLE:  TTIN
;
;       TTIN CHECKS FOR A CHARACTER IN THE TT BUFFER
;       GRABS IT, STRIPS THE ASCII CARRIER (200), AND
;       RETURNS THE CHARACTER IN R0.
;
TTIN:   BIT     #200,@#RCSR             ;CHECK FOR BUFFER FULL
        BEQ     TTIN                    ;NO, WAIT
        MOVB    @#DATIN,R0              ;YES, PUT DATA IN R0
        BIC     #177600,R0              ;STRIP ASCII CARRIER
        RTS     7                       ;GO BACK
;
;       SUBR TITLE:  TTOUT
;
;       TTOUT CHECKS FOR AN EMPTY TT BUFFER AND WRITES
;       OUT WHAT HAS BEEN INPUT
;
TTOUT:  MOV     R0,@#DATOUT             ;PRINT DATA
TT2:    BIT     #200,@#PCSR             ;CHECK FOR BUFFER READY
        BEQ     TT2                     ;NO,WAIT
        RTS     7
        PCSR=177564
        DATOUT=177566
        RCSR=177560
        DATIN=177562
;       SUBR TITLE: WRITE
;       WRITES THE DECIMAL CHARACTERS FOUND IN R1
;
DECURT: MOV     R1,NUMBER               ;GET BINARY NUMBER
        MOV     #COLL,R2                ;GET COLLUM ADRESS
        MOV     #DATA,R3                ;STORE BUFFER ADRESS
        BR      BD2                     ;START CONVERSION
BD0:    INC     CHAR                    ;INCREMENT ASCII CHAR
BD1:    SUB     (R2),NUMBER             ;REDUCE NUMBER
        BCC     BD0                     ;OVERFLOW?
        ADD     (R2)+,NUMBER            ;RESTORE NUMBER
        MOVB    CHAR,(R3)+              ;PUT ASCII CHAR IN BUFFER
BD2:    MOV     #60,CHAR                ;SET CHAR TO ASCII 0
        TST     (R2)                    ;ALL DONE
        BGT     BD1                     ;NO! KEEP GOING
        ADD     NUMBER,CHAR             ;DO UNITS
        MOVB    CHAR,(R3)+              ;PUT LAST CHAR IN BUFFER
        MOV     #DATA,R3                ;GET BUFFER ADRESS AGAIN
PRTOUT: MOV     #5,R2                   ;SET UP COUNTER
ADP1:   BIT     #200,@#PCSR             ;WAIT FOR PRINTER
        BEQ     ADP1                    ;WAIT LOOP
        MOVB    (R3)+,@#DATOUT          ;PRINT CHARACTER
        DEC     R2                      ;BUMP COUNTER
        BEQ     OUT1                    ;DONE. GET OUT
        BR      ADP1                    ;PRINT NEXT CHARACTER
OUT1:   RTS     7                               ;GET OUT
NUMBER: .WORD   0
CHAR:   .WORD   0
PRTSAV: .WORD 0
DATA:   .BLKW   12
COLL:   .WORD   10000.
```

Fig. 3 - Assembly Language Listing of Program to Multiply Two Numbers

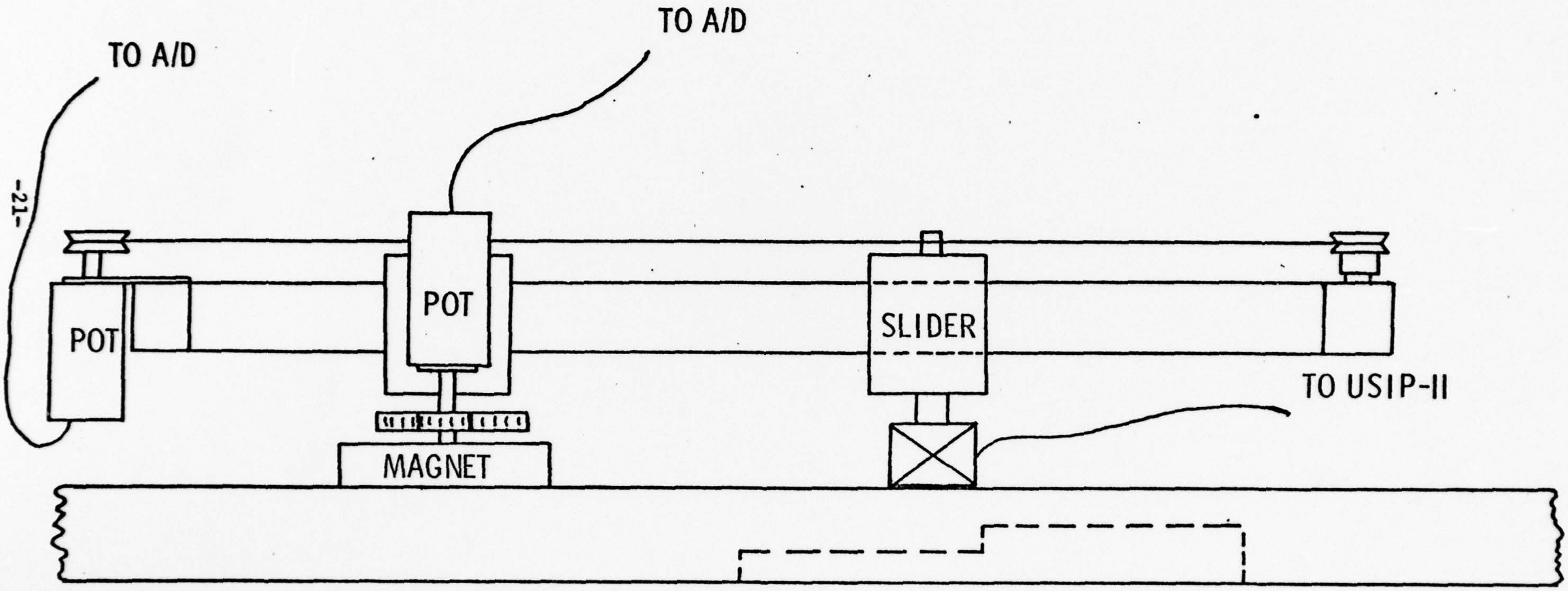



Fig. 4 - SHIP HULL PLATE SCANNER

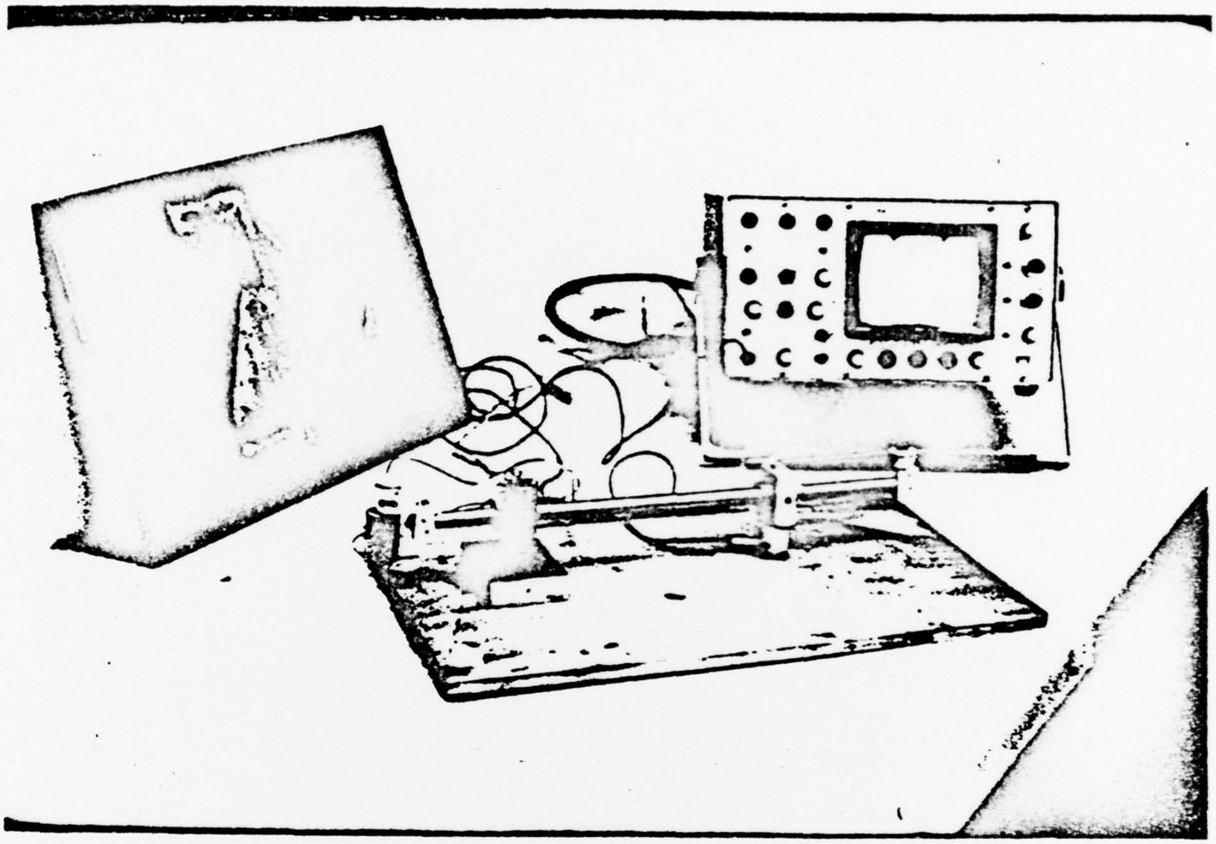


Fig. 5 - C-scan Data Acquisition System

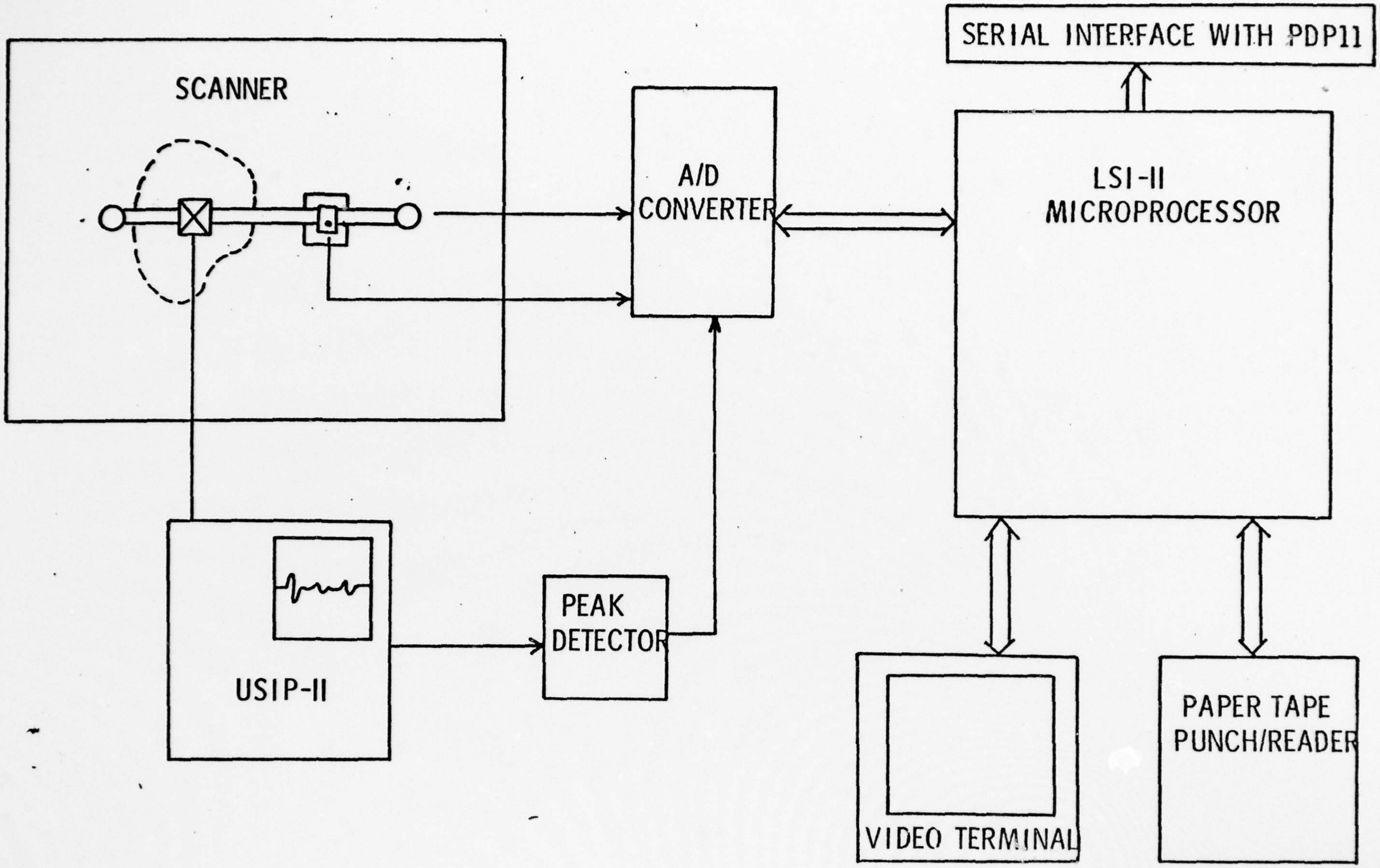



Fig. 6- SHIP HULL INSPECTION SYSTEM

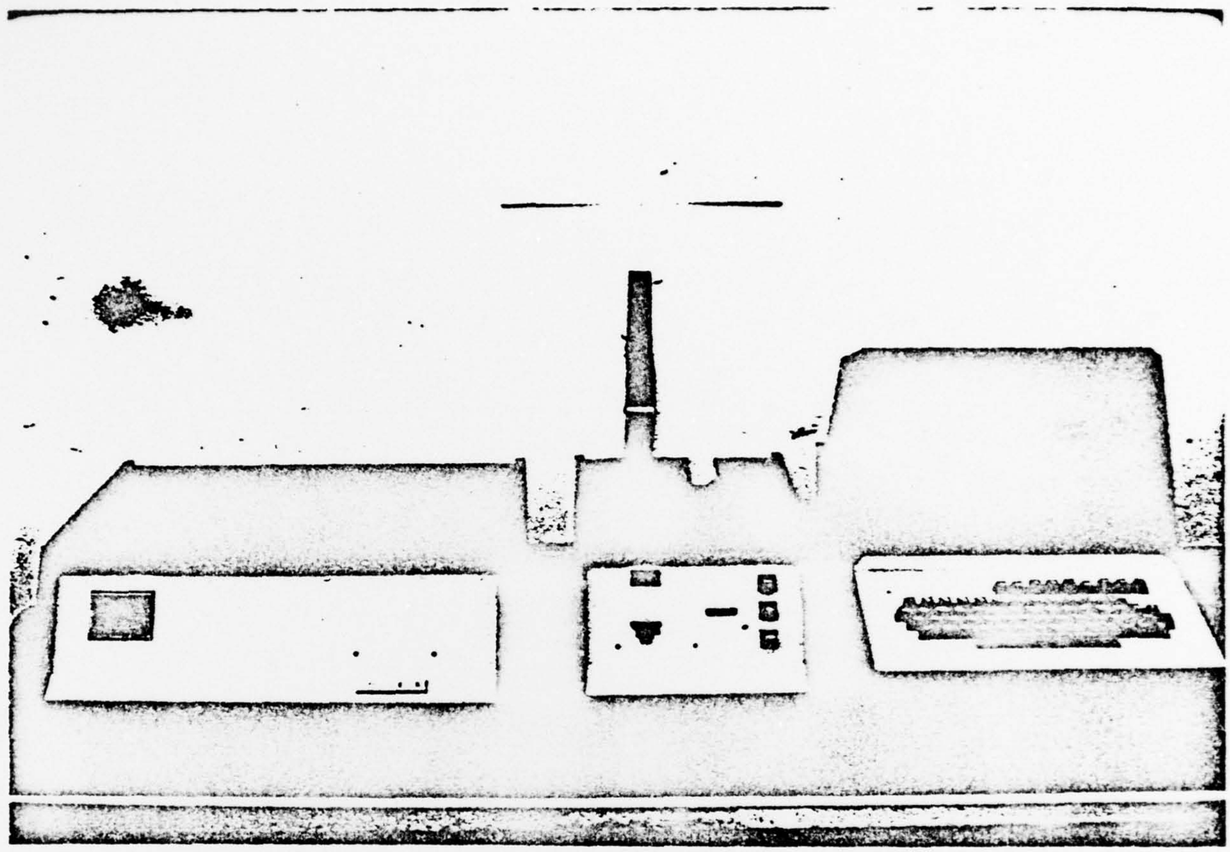


Fig. 7 - LSI-II Microprocessor System: Central Processing Unit, Paper Tape Reader/punch, and Video Terminal

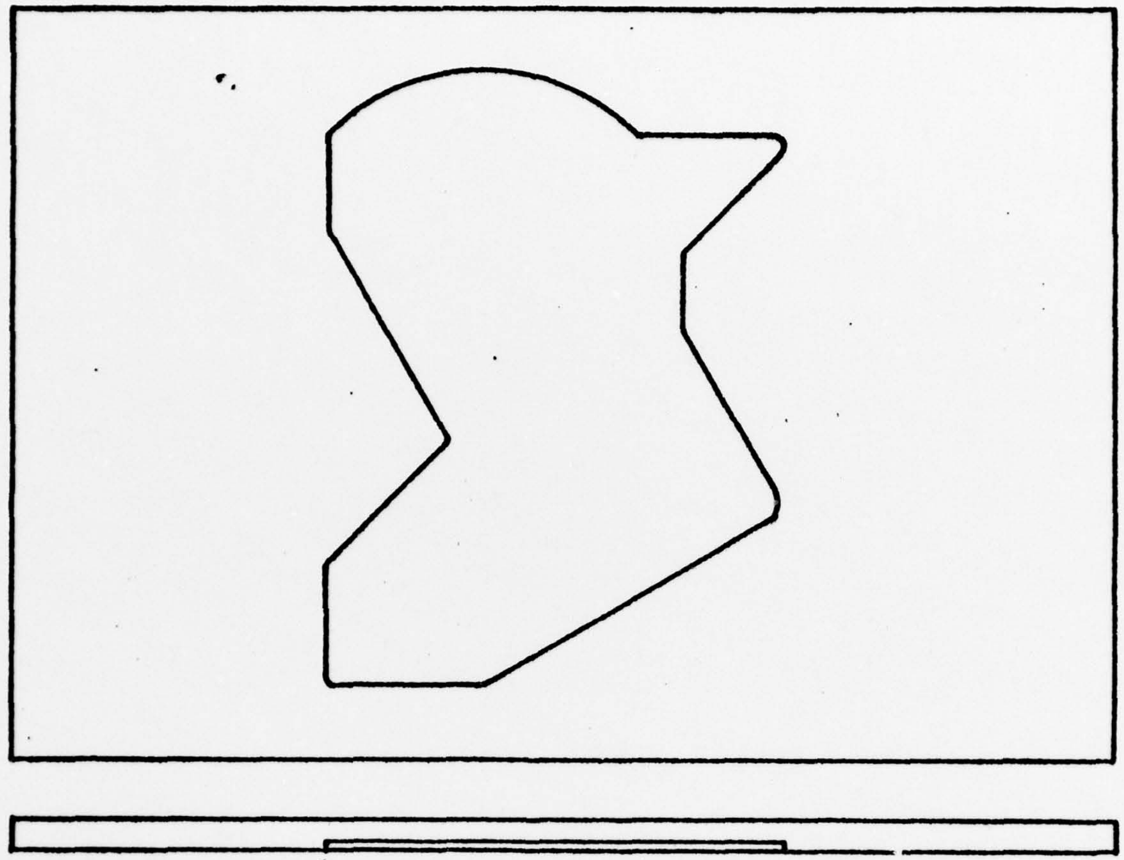


Fig. 8 - ULTRASONIC TEST SPECIMEN TO MODEL DEFECTIVE SHIP HULL PLATE ( MODEL 1 )

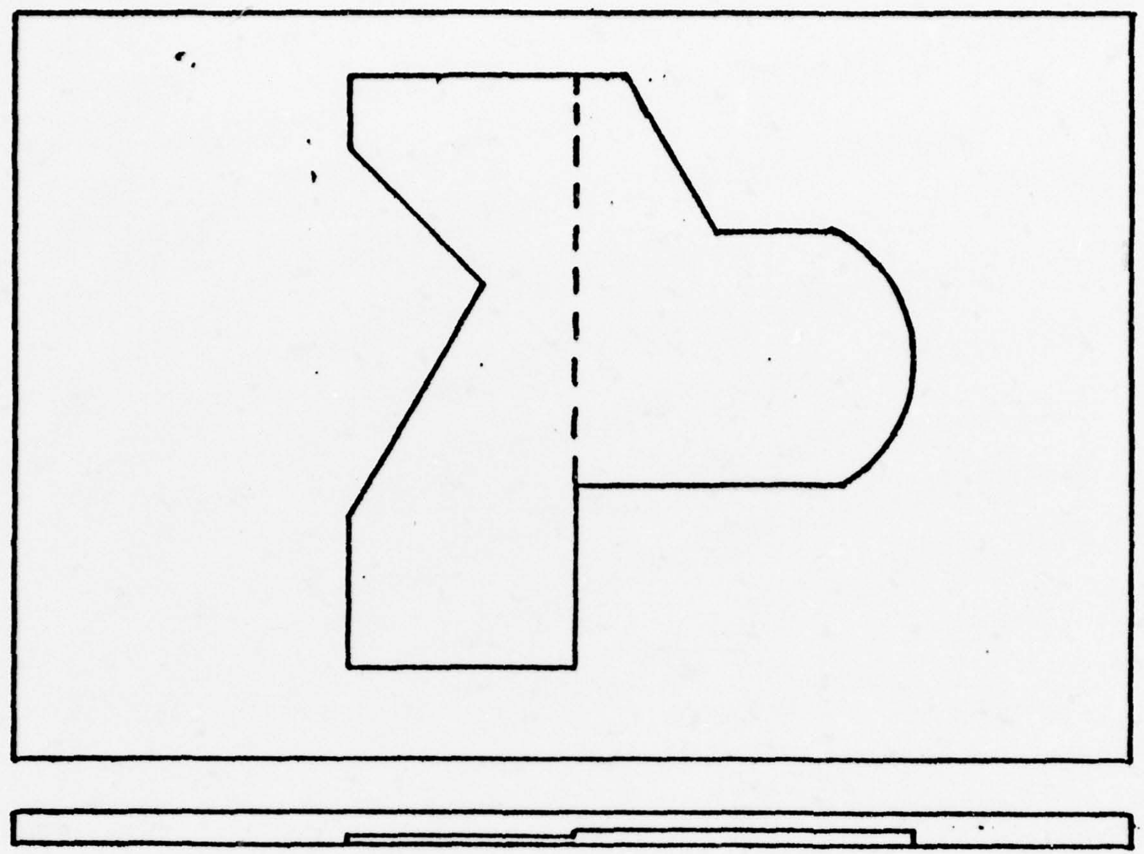


Fig. 9 - ULTRASONIC TEST SPECIMEN TO MODEL DEFECTIVE SHIP HULL PLATE - TWO DIFFERENT DEPTHS ( MODEL 2 )

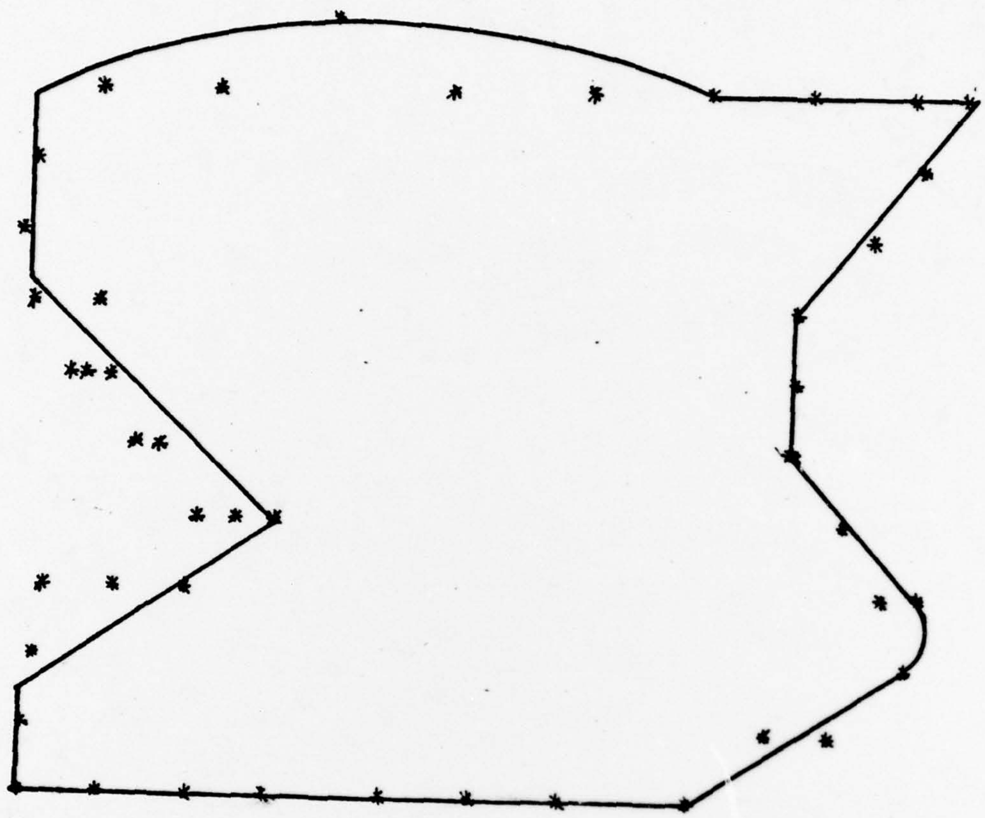


Fig. 10 - TYPICAL EXAMPLE OF THE GRAPHICS DISPLAY SHOWING A FLAW MAPPED BY THE C-SCAN SHIP HULL INSPECTION SYSTEM (MODEL I)

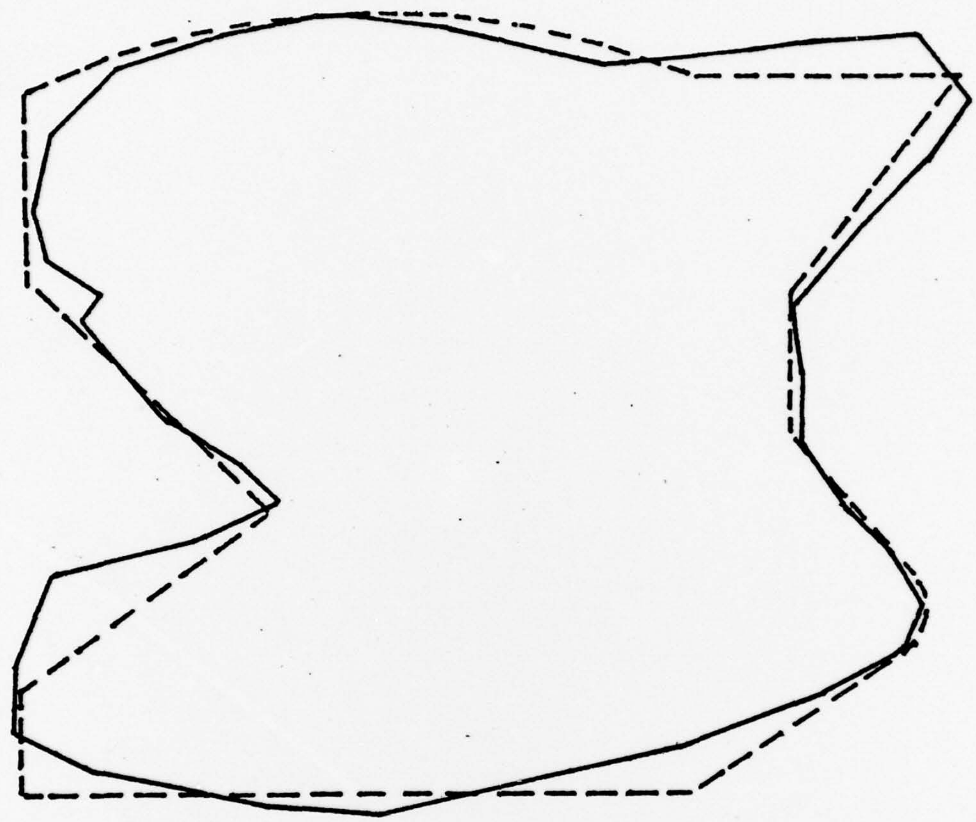


Fig. 11 - PLOT OF FLAW FROM PDP11 TERMINAL USING C-SCAN COORDINATES (MODEL I)

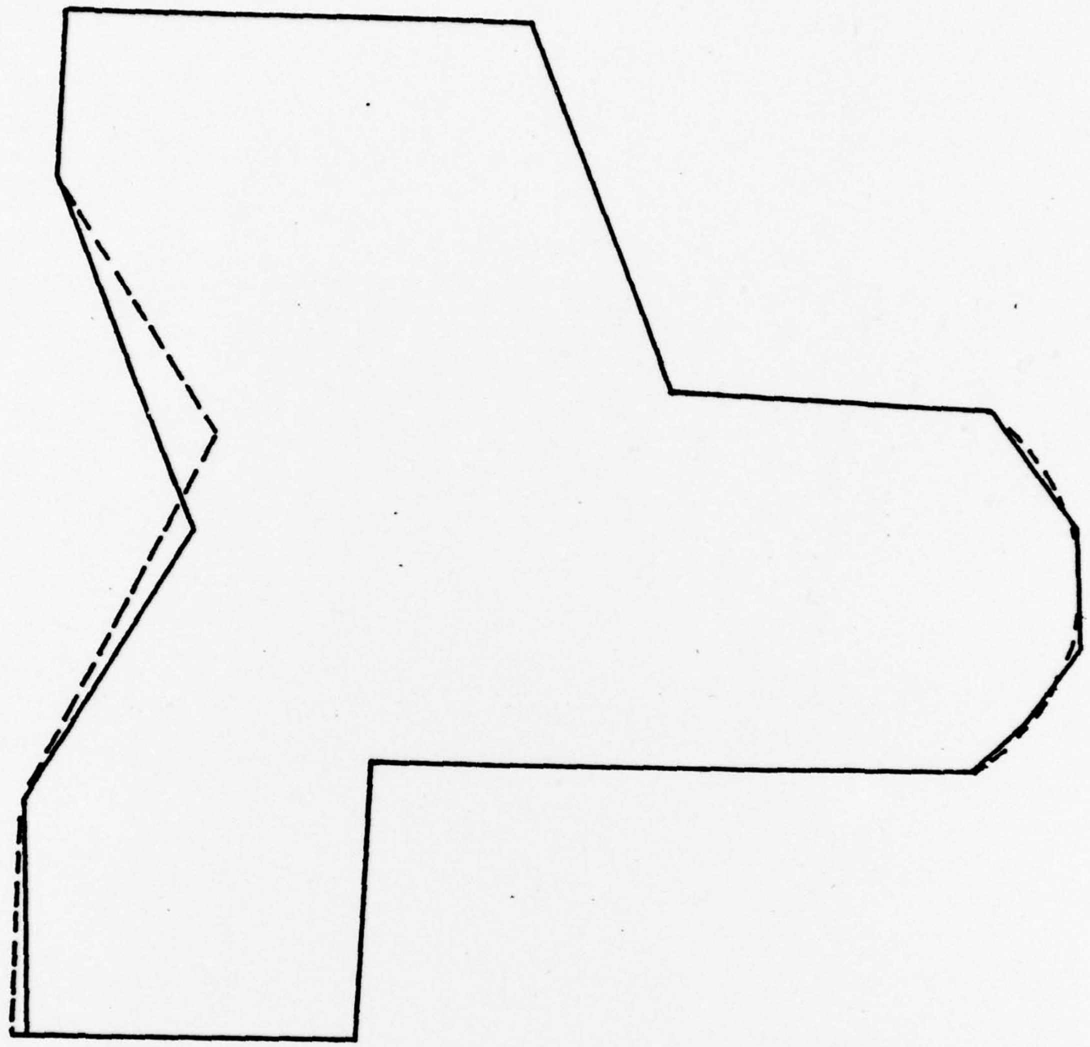


Fig. 12 - PLOT OF FLAW FROM PDPII TERMINAL USING C-SCAN COORDINATES (MODEL 2)

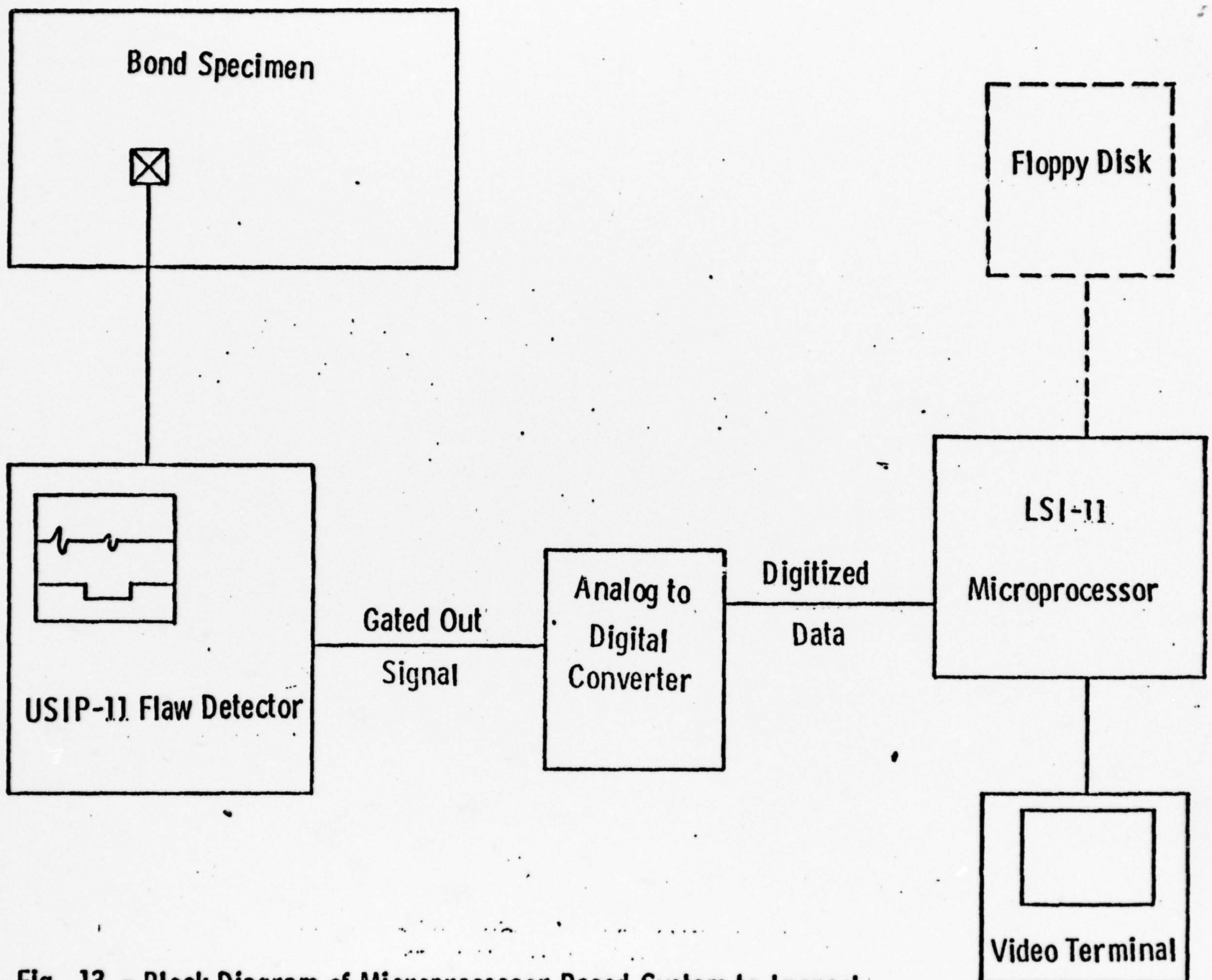



Fig. 13 - Block Diagram of Microprocessor Based System to Inspect Adhesive Bonds

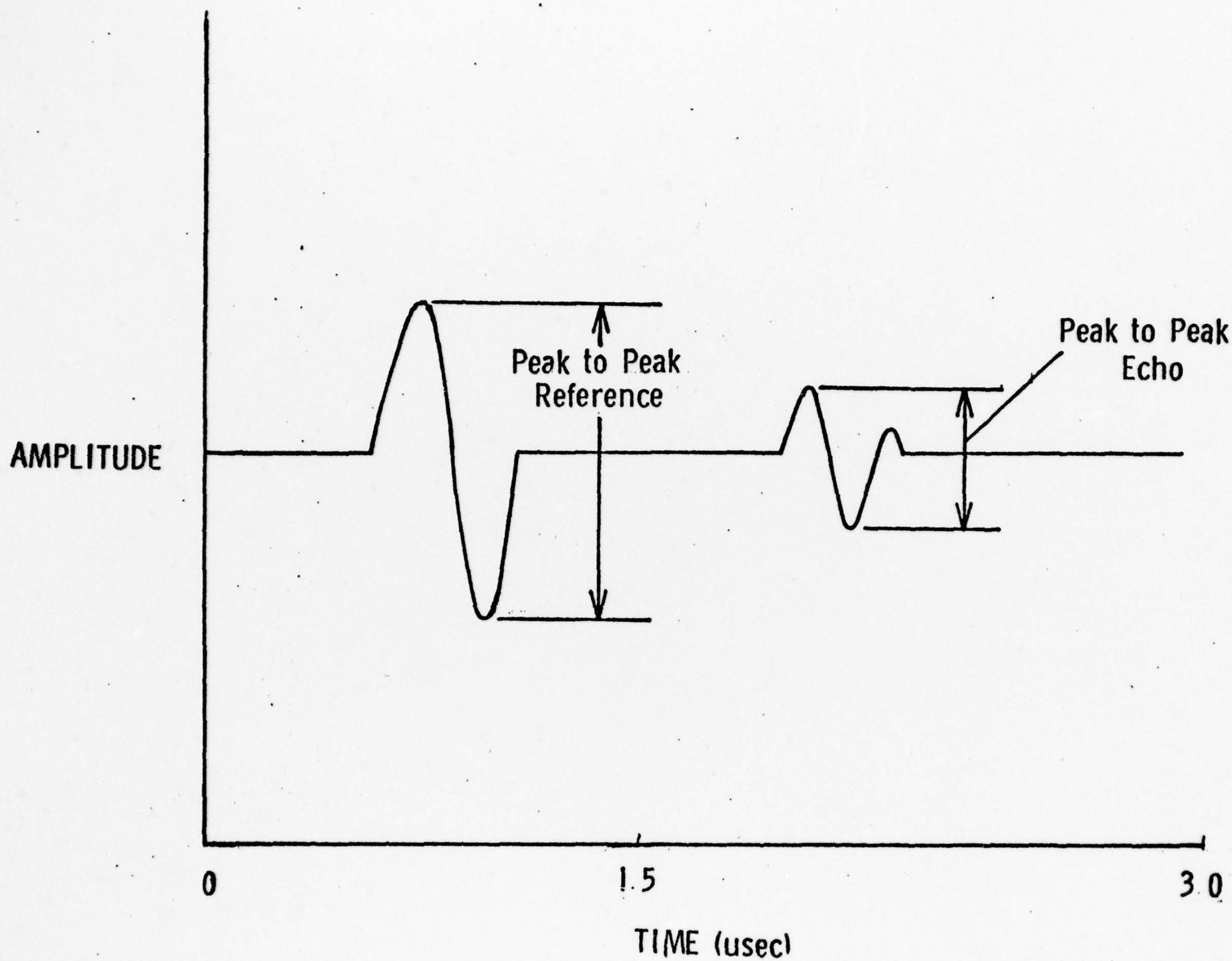



Fig. 14 - Typical Ultrasonic Signal Reflected From Bond Specimen, Displaying Reference And Bond Layer Echoes.

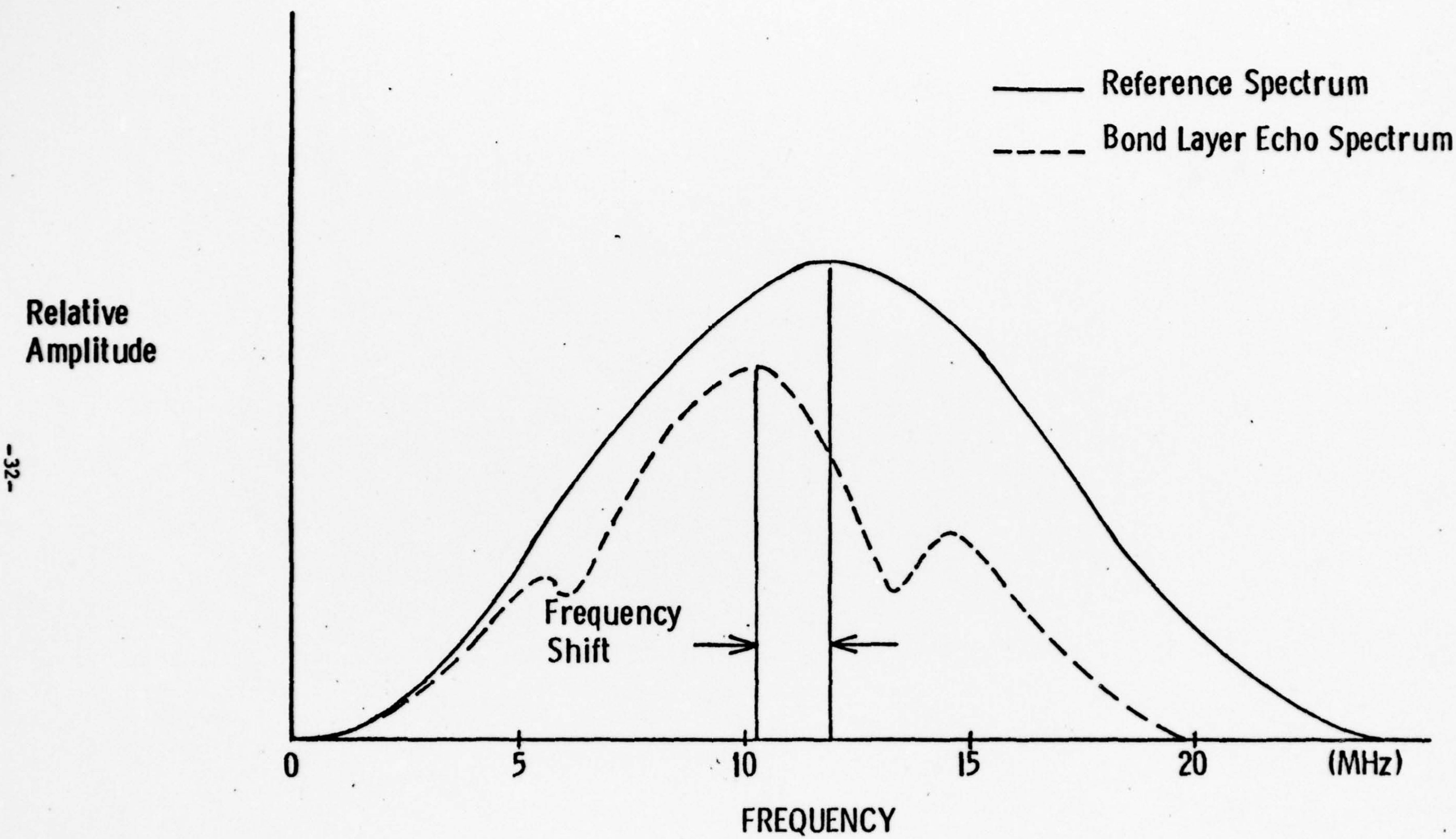



Fig. 15 - Feature Selection From Fourier Spectrum Domain

A - Peak Frequency
B - Dip Frequency
C - Dip - Peak Frequency
D - Dip/Peak Amplitude
E - DP2, - DPI Frequency
F - 6 db Down Bandwidth

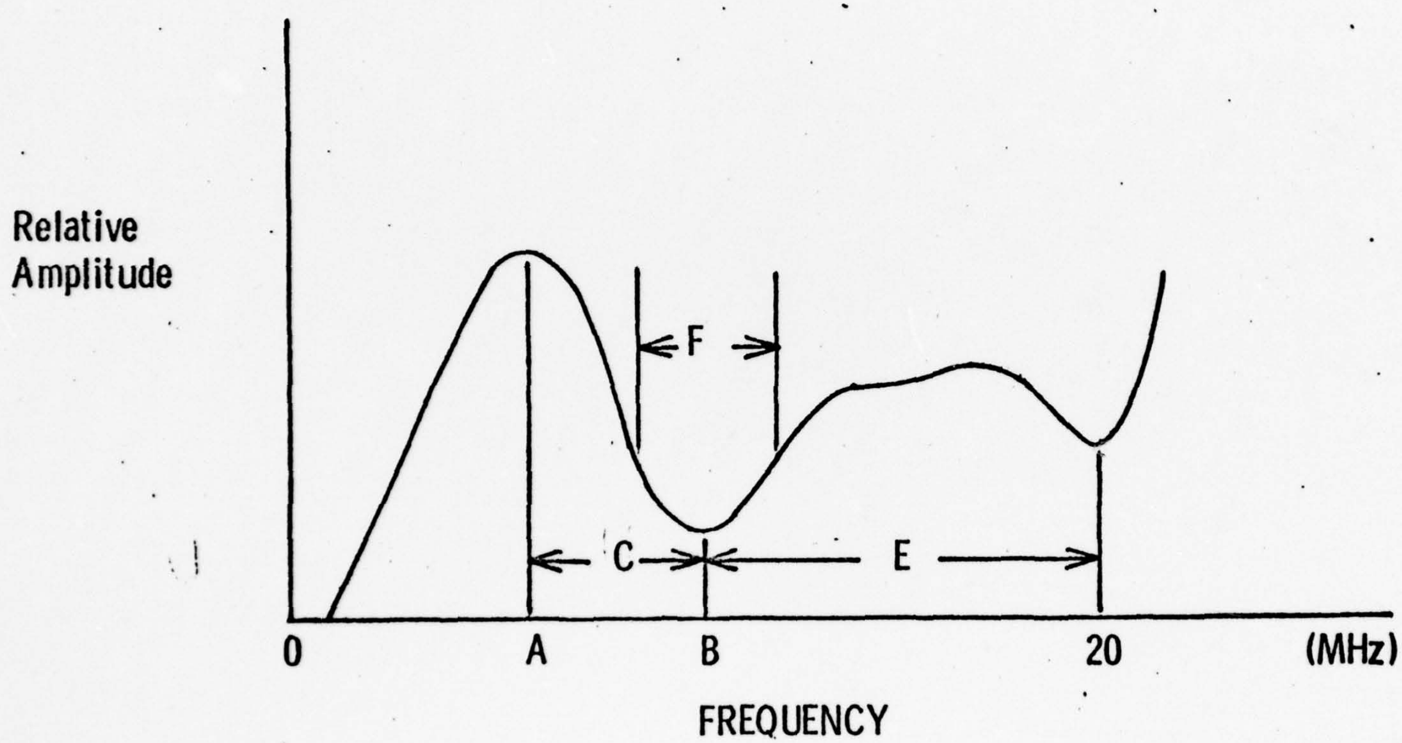



Fig. 16 - Features Selected From Transfer Function

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

| | |
|---|---|
| 1. REPORT NUMBER: N00014-77-C-0607 (4) | 2. GOVT ACCESSION NO. |
| 3. RECIPIENT'S CATALOG NUMBER | |
| 4. TITLE (and Subtitle): Microprocessor Utilization in Ultrasonic Inspection Flaw Growth Classification Studies | 5. TYPE OF REPORT & PERIOD COVERED: Technical Report |
| | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s): Joseph L. Rose, Graham H. Thomas | 8. CONTRACT OR GRANT NUMBER(s): N00014-77-C-0607 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS: Drexel University, 32nd & Chestnut Streets, Philadelphia, PA 19104 | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NR 064-583 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS: Department of Navy, Office of Naval Research, Structural Mechanics Program, Arlington, VA 22217 | 12. REPORT DATE: June 15, 1979 |
| | 13. NUMBER OF PAGES: 36 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) | 15. SECURITY CLASS. (of this report): Unclassified |
| | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release, Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Microprocessor
C-Scan
Ultrasonics
Minicomputer
Nondestructive Testing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The purpose of this work is to demonstrate the feasibility of using microprocessors in ultrasonic inspection. Benefits of employing a microprocessor are two-fold, the first with respect to the development of a small, compact and portable inspection device, and the second with respect to obtaining flaw data in digital format for use in structural mechanics programs for stress analysis. An ultrasonic problem was selected that clearly demonstrates the capability of a microprocessor with initial attention on problem simplicity from an ultrasonic

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

inspection point of view. Such items as data storage, computational speed, computer language selection and utilization, interfacing techniques and the use of such peripheral devices as analog to digital converters and display devices are reviewed in the paper.

The problem selected for this study is that of modelling an ultrasonic inspection of the steel plates from the hull of a ship. The procedure will be applicable to both quality control during manufacture and in-service inspection. Procedures for implementing an advanced ultrasonic inspection algorithm applicable to adhesive bond inspection is also outlined in the